13949

Review of Cognitive Architectures for the design, test, and evolution of intelligent ground vehicles

F. Mili,
Oakland University,

C. Adams, S. Dungrani
US ARMY TARDEC

Abstract

Human behavioral models are computer programs that emulate humans performing tasks and executing missions. Human cognitive models are behavioral models with a focus on cognitive tasks involving planning, problem solving, and learning among other things. A quarter century of research in Artificial Intelligence, Cognitive Science, Computer Human Interaction, and Computer Simulation has generated a rich collection of competing theories and tools for developing cognitive models. While the field of human modeling theories and tools continues to grow, and the awareness of the potential benefits of human modeling increases, there is very still a lack of comprehensive, unbiased guidelines allowing engineers to compare tools and select those appropriate for the task at hand. Two reports have been commissioned in the past 5 years by the Defense Modeling and Simulation Office, and the HSIAC respectively. The work reported in this paper shares the same goals and motivations as these two books, albeit with a narrower and more focused scope. Specifically, this paper examines the applicability of existing cognitive architectures as they apply to the design, test, and evolution of intelligent ground vehicles.

1. Introduction

Human behavioral models are computer programs that emulate humans performing tasks and executing missions. Human cognitive models are behavioral models with a focus on cognitive tasks involving planning, problem solving, and learning among other things. A quarter century of research in Artificial Intelligence, Cognitive Science, Computer Human Interaction, and Computer Simulation has generated a rich collection of competing theories and tools for developing cognitive models. Work in artificial intelligence has produced "perfectly intelligent" systems able to use their knowledge to its fullest in order to prove theorems, create optimal plans, interpret images, and perform other tasks generally thought of as requiring intelligence [Sim96]. The mission of Cognitive science research has been "to find the order that exists in the complexity" of the human mind [AnL98], i.e. to construct a theory that explains the wealth of qualitative and quantitative data accumulated by psychological studies and neurophysiological methods on how humans memorize, recall, and use facts [AlM01]. After years of attempting to explain these human cognitive "regularities" individually by developing partial theories, Alan Newell triggered a major shift in approach in his famous "You can't

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
09 SEP 2003	**Journal Article**	**08-08-2003 to 08-09-2003**

4. TITLE AND SUBTITLE
Review of Cognitive Architectures for the design, test, and evolution of intelligent ground vehicles

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
F Mill; S Dungrani; C Adams

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Oakland University,2200 North Squirrel Road,Rochester,Mi,48309

8. PERFORMING ORGANIZATION REPORT NUMBER
; #13949

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army TARDEC, 6501 East Eleven Mile Rd, Warren, Mi, 48397-5000

10. SPONSOR/MONITOR’S ACRONYM(S)
TARDEC

11. SPONSOR/MONITOR’S REPORT NUMBER(S)
#13949

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Human behavioral models are computer programs that emulate humans performing tasks and executing missions. Human cognitive models are behavioral models with a focus on cognitive tasks involving planning, problem solving, and learning among other things. A quarter century of research in Artificial Intelligence, Cognitive Science, Computer Human Interaction, and Computer Simulation has generated a rich collection of competing theories and tools for developing cognitive models. While the field of human modeling theories and tools continues to grow, and the awareness of the potential benefits of human modeling increases, there is very still a lack of comprehensive, unbiased guidelines allowing engineers to compare tools and select those appropriate for the task at hand. Two reports have been commissioned in the past 5 years by the Defense Modeling and Simulation Office, and the HSIAC respectively. The work reported in this paper shares the same goals and motivations as these two books, albeit with a narrower and more focused scope. Specifically, this paper examines the applicability of existing cognitive architectures as they apply to the design, test, and evolution of intelligent ground vehicles.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Public Release**	**12**	

consumer of human behavioral modeling work. In effect, military applications span the whole spectrum of the applications mentioned above: Many military systems are mission-critical or safety critical, or both; the major activity during peace time in the military is training, with computer-based simulation and tutors playing an important role; and synthetic agents and environments play a critical role in developing and testing new doctrines, and in training personnel.

While the field of human modeling theories and tools continues to grow, and the awareness of the potential benefits of human modeling increases, there is very still a lack of comprehensive, unbiased guidelines allowing engineers to compare tools and select those appropriate for the task at hand. Identifying the right tool or combination of tools for the task at hand is increasingly difficult. There are reports of experimental comparisons between the major competing theories (Soar, ACT-R, EPIC) in the context of specific, small scale applications of planning [], game playing [], and multi-tasking[Salv01]. The (only) most comprehensive studies have, in fact, been commissioned by the military. The first study was commissioned the Defense Modeling and Simulation Office. A panel on Modeling Human Behavior and Command Decision Making conducted a two-year study and summarized its findings in a book edited by R.W. Pew and A.S. Mavor [PeM98]. In this book, the authors establish criteria for evaluating the modeling tools; they summarize their comparative evaluation of the different tools in a table with tools ranging from cognitive architectures to performance simulation tools. The panel concluded that (1) the accurate modeling of human behavior is essential to the success of military applications, that (2) military applications are not always using the latest research, and that (3) there is still great potential in using the most advanced science in human modeling, and in contributing to that science. They recommend increased funding of research in human behavioral modeling and increased collaboration with the research community. A second report was commissioned and published by the Human Systems Information Analysis Center (HSIAC) [Rit03]. This later report updates the Pew and Mavor book by reporting on more recent developments, notably on research done in European centers, and identifies additional research issues for the human behavioral modeling community.

The work reported in this paper shares the same goals and motivations as the two above-mentioned efforts. It is a obviously of a much smaller scope than the multi-person multi-year endeavors commissioned by Defense Modeling and Simulation Office and by the HSIAC. On the other hand, it has a narrower and more focused scope: The use of existing cognitive architectures for the design, test, and evolution of intelligent ground vehicles. An upcoming report will address engineering performance models for intelligent ground vehicles. This paper is organized as follows: In section 2, we provide background information, definitions, and terminology. In section 3, we discuss the requirements for the domain of intelligent ground vehicles. In section 4 we review the major cognitive architectures. We compare them with respect to functional requirements in section 5 and with respect to non functional requirements in section 6. We summarize and conclude in section 5.

architecture. They come in the form of storage and processing structures (e.g. long term memory, visual buffer, etc.), algorithms specifying how the contents of these structures are used and changed, and constraints on the structures and processes (e.g. timing constraints, concurrency constraints).

- ***The task-specific and environment specific aspects.*** This is the knowledge-level description of the task including a description of the goals, a description of the environment, and a definition of the processes (e.g. productions) that can be used to reach the goal. According to Newell, "if humans were perfect intelligence systems", this is the only description that we need to model task execution behavior.
- ***The behavioral moderating aspects.*** Because behavior and performance vary from person to person on the same task, and vary for the same person same task depending on the situation, it is important to reflect this variability. The variability can be captured through a randomization process to reflect a more or less accurate distribution of human variability. It can also be captured via parameter setting, and by accounting for past experience.

It is possible to develop a cognitive model, in an ad hoc manner using any programming language. For the sake of validity and efficiency, it is much easier to develop cognitive models using existing architectures and using existing architecture-specific simulation systems. Most cognitive architectures have been materialized in implemented systems. A cognitive architecture system is a software package with the architecture's structures and processes built into it. Figure 1 below illustrates the different components

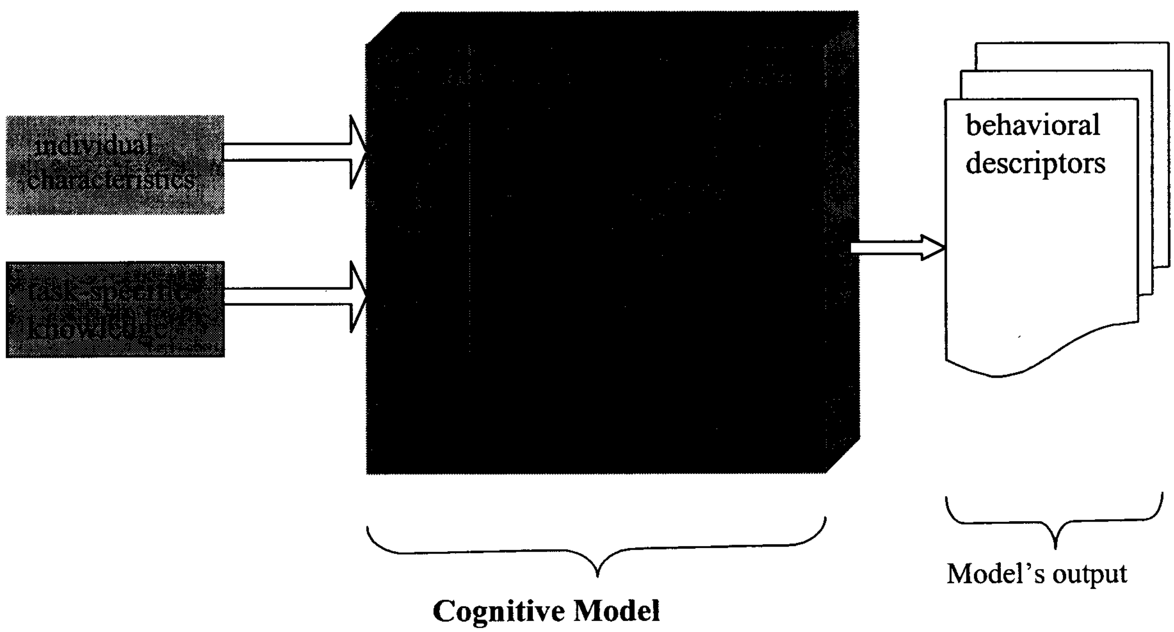

Figure 1

3.1 Introduction

Intelligent Ground Vehicles are manned or unmanned vehicles with some generic functions and specialized features [Ada03]. Manned vehicles have a crew of varying size. While manned or not, these vehicles can have different platforms. IGVs are used to execute missions. They are generally part of a larger organizational unit; they are assigned a role, and are expected to execute their mission, and to communicate with other IGVs that can be peers, superiors, or subordinates. Because of the need to communicate with other vehicles, an IGV behavior must be "manned-like" whether it is manned or not. In other words, it is important for other IGVs communicating with it not to be able to distinguish between manned and unmanned IGVs.

3.2 Applications

Cognitive modeling can be used in the full spectrum of applications within the context of IGV: research and development, testing, training, and development of synthetic forces. These applications invoke two different scenarios: analysis and automation. In the analysis application, human cognitive models are developed and used to collect information about the way the overall system crew-IGV is likely to function. This is illustrated in Figure 2. In this figure, the research/developer actor examines the process of executing a task by the IGV-crew in various environments and identify areas of vulnerability as well as areas of opportunities. By materializing the crew with cognitive modes, extensive tests can be run to weaknesses in the IGV-crew system, sources of overload, stress, and errors.

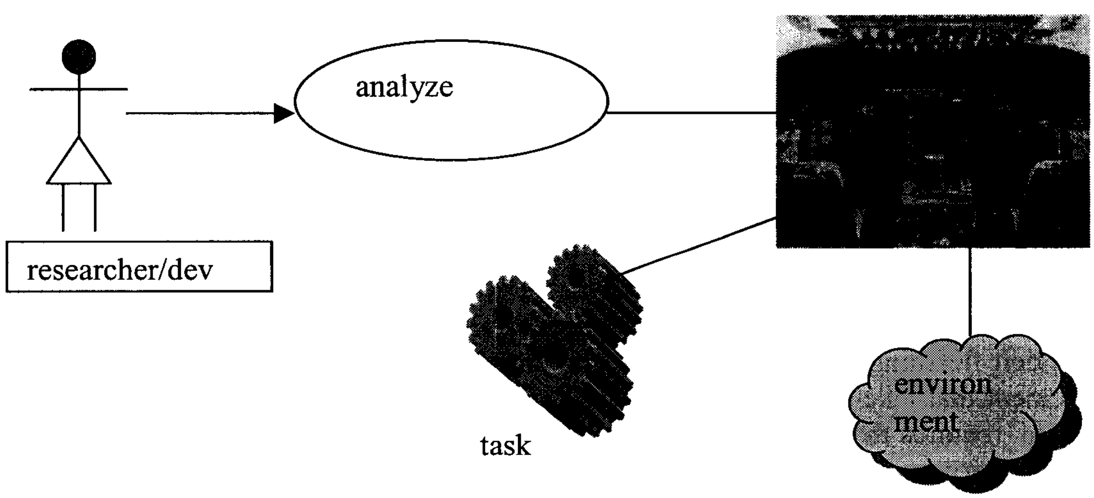

Figure 2

In automation, selected tasks or all the tasks are automated by creating a synthetic crew to receive instructions from superiors, execute missions, communicate with other IGVs, and answer questions concerning status, progress, and rationale for decisions made. A prototypical example of this level of automation in the aviation domain is the TacAir Soar system developed by Nielson et al [Nie02] from SoarTech. The TacAir Soar

Emotion:

Albus and Meystel argue that intelligent behavior is driven by two types of signals, internal and external. The external signals are the outside environment as perceived by the intelligent agent; the internal signals comprise basic drives such as thirst, hunger, but also emotions. Emotions and other internal behavioral moderators are increasingly recognized as important aspects of human cognitive behavior that must be accounted for by the cognitive models and the underlying architectures. Fear for example, is often interpreted as an optimized mechanism to react quickly to life threatening situation even under situations with information overload and ambiguous input [Wha99]. In their book *The Cognitive Structure of Emotion,* Ortony *et al* [Ort98] identify twenty two distinct types of positive and negative emotions including happiness, fear, resentment, anger, and remorse. For the IGV context, behavioral moderators must be accounted for during analysis and testing in order to get an accurate picture of crew's behavior under stress, fear, fatigue, sleep deprivation, and other factors. In the case of automation, it is also important for the synthetic crew to behave in a human-like manner in its interactions with other vehicles and to account for the potential effects of these moderators on other vehicles it is interacting with. Ritter et al [Rit03] emphasize the need to model behavioral moderators to be able to model "non-doctrinal" behavior such as insubordination and erroneous behavior.

Errors:

To be complete, cognitive models must account for human cognitive behavior, in all of its aspects, including erroneous decisions and behavior. Modeling with the assumption that human behavior is normative produce results that are not accurate. In particular, modeling used to test interfaces and analyze performance of the crew while using a specific platform is most informative when it identifies the most likely errors that can be made by the crew. This information is a valuable resource for reviewing the design and revising it in order to minimize these errors and in order to learn how to recover from these errors in a safe way. The modeling of errors is less important for synthetic crew because a key advantage of automation is an improved performance. On the other hand, as pointed out by Ritter et al [Rit03], it is important to account for errors by modeling recovery from them.

Variability:

Developing models that perform a task in a non-repeatable way introduce an additional element of accuracy especially for automation. Variability in behavior is not as much needed for routine normative tasks, but is important when multiple approaches and strategies are possible. Variability is also important in adversarial situations when it is important not to behave in a too predictable way.

Multi-tasking:

Early efforts in cognitive modeling focused their efforts on capturing the problem solving aspects of cognition, which by nature, are attention focused and goal oriented. In practice most situations (e.g. driving, flying an airplane) consist of balancing between multiple tasks, assigning limited resources to them in such a way that they all get accomplished. Multi-tasking raises issues of scheduling and resource allocation to competing goals. It contribute something unique and take an additive approach, i.e. are compatible with other architectures.

4.1 ACT-R (Adaptive Control of Thought)

"The challenge of science is to find the order that exists in the complexity of the world."
Anderson and Labiere, The Atomic Components of Thought, 1998.

The name ACT-R is used for both the cognitive architecture and the implemented system. ACT-R is one of the earliest (and still on-going) efforts in cognitive modeling led by Anderson et al from Carnegie Mellon University. ACT-R has evolved out of HAM, a theory of human declarative memory developed in 1973 by Anderson and Bower and emerged under the name ACT-E as an integration of the memory model embodied by HAM and the model of how the knowledge in memory is deployed embodied by a production system [AnL98].
One of the key characteristics of ACT-R is its distinction between the symbolic and the sub-symbolic level. Facts and inferences are expressed at the symbolic level using explicit structures. Relevance of these facts and their vividness in memory (thus speed and accuracy of their recall) are expressed at the sub-symbolic level through numerical functions. The nature of the numerical functions capturing the sub-symbolic level "knowledge" have been evolving through the years from neural-net like structures in ACT* (1978) to statistical functions reflecting Anderson's rational theory of decision making in ACT-R (1993).
ACT-R has continued to evolve in terms of its underlying theory as well as in terms of implementation. Among the notable evolutions of the theory and implementation is the integration of perception and action. Because the original ACT-R architecture has been focusing on higher cognitive functions, it has one cognitive processor along with its buffers. Yet, cognition does not happen in a vacuum, it is triggered by and affected by visual, auditive, and tactile perception and is reflected in terms of vocal and motor actions. It is now widely believed that, to be accurate, cognitive modeling must be integrated with perception and action []. ACT-R has adopted ideas and parameters from the multi-modal EPIC architecture, thus adding perception and action processors along with their associated buffers, performance parameters, and processes. These features are part of the ACRT-R/PM system.

The main key features of ACT-R are:

- It distinguishes between declarative memory containing facts (e.g. 3+5=8) and process memory containing production (e.g. to add two numbers, add the units first, ...). These distinctions have been motivated and supported by psychological and medical evidence to the fact that declarative and procedural knowledge are stored in separate location in the cortex [].
- Chunks and productions are considered to be the **atoms** of cognition. In other words, while modeling, facts and productions must be decomposed to ensure that they are all expressed at the atomic level. Composite facts are not compatible with the architecture's assumptions.
- ACT-R distinguishes between symbolic level (3+5=8) and sub-symbolic level (speed and accuracy of recalling that 3+5=8). In other words, facts and

- Soar allows the modeling of problem solving at different levels. The grain size of the output reflects the grain size of knowledge. This is in contrast with ACT-R where all facts and productions must be captured at the atomic (50ms) level.

- Soar has been interfaced seamlessly with EPIC.
- Soar uses a two-level control structure that distinguishes between the automatic access to the knowledge (all relevant productions are recognized) and the deliberate problem solving where a "decision" is selected and executed. Each cognitive step in Soar is accomplished by the so-called *recognize-decide-act* cycle. An impasse is reached if the recognize cycle produces no alternatives or if the decide cycle does not have enough information to decide.

Soar has been evolving both as a theory and as an implementation since its initial introduction. The recognize-decide-act for example is the result of a number of refinements motivated by the need to explain deliberate problem solving. This led to the introduction of symbolic preferences to replace the voting mechanism used in Soar1.
The first publicly available, fully documented version is Soar 4.
One of the major concerns of Soar's implementation has been efficiency, scalability, and portability. This is reflected in its choices of programming language (switched from Lisp to C) and indexing and retrieval mechanism (Rete algorithm).

4.3 EPIC

"Assimilating the fundamentals of contemporary computer OSs into theories of cognitive control opens many promising paths for future research. With this assimilation, it will be possible to characterize a wider range of control functions more precisely, and to test more definitively for the existence of general as well as customized executive processes. These advances also will lead to more detailed and veridical analyses of multitasking skill acquisition. Computational Modeling based on the EPIC architecture provides one vehicle whereby this progress can occur."

Kieras et al. 1999

The EPIC cognitive architecture was introduced by Kieras and Meyer in 1995 motivated by the need to accurately model multi-modal processing in time-stressed tasks. The two key characteristics of EPIC are: 1. It integrates cognition with multi-modal perception and action; 2. It is specifically adapted towards multi-tasking. Its name Executive Process Integrative Control reflects the fact that it uses an executive process to manage resources among multiple concurrent tasks. EPIC is a more recent addition to the pool of candidate cognitive architectures. It answers the need for a comprehensive computational theory of multiple-task performance that allows quantitative prediction of mental workloads in multi-tasking situations. EPIC has evolved out of CPM-GOMS, a member of the GOMS family of methodologies used to evaluate human computer interfaces.
One important way in which EPIC departs from ACT-R and Soar is in terms of human performance bottleneck. ACT-R and Soar assume that human performance is bound by the fact that the cognitive processor is essentially sequential. EPIC developers challenge this hypothesis which does not hold in multi-tasking situations. For multi-modal, multi-

- In each decision cycle, the contents of the working memory are matched with the productions. All productions who match with a given level are fired within the same cycle. There is no conflict resolution.
- Each working memory element has an activation level. These activation levels are changed by the firing productions.
- Elements in the working memory must have their activation level beyond a given threshold to match productions.
- 3CAPS places a ceiling to the total activation level of its working memory elements at any given time. When the level is exceeded, all activation levels are scaled down. This process models (short) memory decay and the effect of overload.
- 3CAPS goals are also stored in working memory with an associated activation level. There is no separate stack structure for goals.

4.5 Cognet

Cognet (COGnition as a NEtwork of Tasks) is a framework for creating and running models of humans executing multiple tasks concurrently. Like EPIC, Cognet is focused on multi-tasking. Unlike EPIC, Cognet is focused on tasks that are mainly cognitive, rather than involving perception and action. Cognet was introduced in 1992 by Zachary et al.

The main characteristics of the Cognet architecture are:

- Humans perform multiple tasks in parallel by switching attention from one task to another.
- Cognet uses the metaphor of the "shrieking demons". Each task is associated with a demon. Demons shriek to get attention. The loudest demon gets attention by having the processor allocated to its task.
- Cognet does not represent the environment explicitly. Perception is captured by a perception process.
- Cognet uses is a blackboard as the common structure and context for negotiation between the different demons..

5. Review of architectures with respect to Functional Requirements.

5.1 Nature of the knowledge

The six architectures presented here are all symbolic architectures based more or less explicitly on Newell's Physical Symbol System theory. Therefore, all six architectures represent knowledge primarily as a collection of productions. There is nevertheless some variability with more or less significant implications.

Productions of the form condition-action or goal-condition-action are an important representation in all of the six architectures.

ACT-R uses production rules to capture the procedural knowledge. ACT-R complements productions with symbolic chunks. A key characteristic of ACT-R's productions is the fact they must be atomic, i.e. they must represent the atom of what can be learned, retrieved, and executed. It cannot potentially encompass other productions. Soar uses

5.5 Learning

ACT-R's architecture incorporates learning both at the symbolic level (new facts and to some extent, new productions) and sub-symbolic level by tuning its parameters to its experience. New facts (and to a lesser extent productions) are learned whenever a new goal is achieved.

Soar learns by chunking. Any time an impasse is created and then solved, a new production is added to the long-term memory. Because Soar captures everything as productions, and because Soar learns new productions, in fact Soar can learn all kinds of productions, productions that propose operators, productions that express preference among productions, and productions that execute operators.

Because of the large number of productions that can be added in this way, learning can be turned on or off in Soar.

The remaining three architectures do not have a learning component.

5.6 Emotion

Given that emotion and other behavioral moderators are seen as part of the major regularities of human behavior, they should be supported within the architecture rather than coded individually in models. None of the five architectures have built in support for behavioral moderators. It is nevertheless possible to encode the behavioral moderators within the models, generally as productions. In ACT-R, the architecture's parameters can be set to modify the behavior in a way consistent with selected emotions. In Cognet, emotions can be modeled as individual agents on their own requesting attention and pre-empting other tasks as needed. To the extent that we can see the size of the working memory as affected by behavioral moderators, 3CAPS supports the expression of moderators. Just and Carpenter claim that a number of differences in proficiency can be explained by differences in working memory size, which shrinks with age. For example, experiments show that elderly people have difficulty repeating sentences with a complex syntax (i.e. sentences that make large demands on working memory).

5.6 Errors

There is a wide variety of reasons why humans make errors. We only mention here the types of errors that are accounted for by the five architectures.

The errors captured by ACT-R are explained in terms of its sub-symbolic level. Facts (chunks) not used recently experience decay and progressively loose accuracy of recall. They take longer to be retrieved and are retrieved with some level of noise. Similarly, preferences for productions over others are increased or decreased according to past experience in using them. As long as current experience is consistent with past experience, the "correct" productions will be selected. On the other hand, the wrong productions will be selected if the current situation differs from past experience.

The architecture 3CAPS is based around a model of short-term memory that explain errors and mistakes made by human when performing certain tasks (post-completion errors for example). In that respect, 3CAPS is well adapted to capturing a specific class of errors.

productions to specify the tasks in addition to some specification of the environment. Tools supporting this activity are instrumental to its success and viability.

6.1 The existence of thorough documentation of the system

Each of ACT-R and Soar are publicly available with a website distributing the software along with tutorials, technical papers, and other useful resources. Both ACRT-R and Soar are sufficiently mature. The on-line tutorials are sufficiently detailed to allow self-learning. In addition, they both offer on-site annual tutorials and workshops. ACT-R is publicly available http://act-r.psy.cmu.edu/. The latest version is ACT-R5.0 available for Windows and Unix operating systems. The ACT-R environment consists of a set of GUI tools for running, inspecting, and debugging ACT-R models. There is a tutorial composed by 9 units available on-line.

Soar 8.4.5 is publicly available from http://www.eecs.umich.edu/main.html for various platforms. The core system is a highly developed programming environment with user manuals, tutorials, and a number of other resources. The core software has a large number of commands used to step execution and to examine the contents of various structures in working memory. In particular, some of Soar's architectural decisions have been driven by the need for the architecture to explain its decisions including why certain operators were or were not used. Soar's output can also be tuned to displaying various statistics. All Soar implementations come with a Tcl/Tk Soar Interface, a GUI interface between Tcl and Soar. Visual Soar is a development environment for the creation of Soar agents. SGIO is a C++ interface library that allows interaction between Soar and other applications. SoarTech has also a set of tools for debugging Soar Debugger and Documentation http://www.soartech.com/downloads. In addition to the above tools, there are a number of others that are user-contributed. They include ViSoar, SDB, SocketIO, Convert, and C_extensions. See www.eecs.umich.edu/~soar/projects.html for more details.

EPIC is a younger architecture. Although it too has a website with code and documentation, it is less detailed. The EPIC system is in fact a library of Lisp functions. Writing a model in EPIC consists of writing Lisp productions. On the other hand, because EPIC has been integrated within both ACT-R and Soar, this is not such a shortcoming. Cognet and 3CAPS are also more recent architectures with fewer available resources.

6.2 Software engineering support

Both ACT-R and Soar are full-fledge environments available for Windows and Unix platforms. Although they both have some forms for debugging and testing, research and development work is still under way in adding more advanced tools. For example, Software engineering tools for debugging, testing, and validating models. The ACT-R development environment supports testing and stepping through the productions. It also supports the insertion of actions that display arbitrary information for the purpose of debugging and visualization

6.3 Support for modularity and reuse

possibilities. The two reports published within a five year interval reflect this increasing interest in pushing cognitive science research further or faster to accomplish the goals it set for itself. After all, none of the existing cognitive architectures in existence today are unified. None of the cognitive architectures is able yet to explain all of the regularities of human cognitive behavior. In his seminal paper in 1975, Newell listed 92 human cognitive regularity. Cognitive architectures are still converging towards explaining all of them. In part because no cognitive architecture is able yet to capture all the regularities known to Newell in 1975 –let alone all of those known to date, there is a burgeoning of architectures being proposed. This raises the need for some roadmap for users in identifying which of the many architectures is applicable and useful for which purpose. This paper is a contribution to building such a roadmap.

We have examined five cognitive architectures and discussed them in light of the functional and non functional requirements of IGV analysis, design, and automation. Whereas our interest is primarily in this application, this review provides insight of general use in the sense that it objectively highlighted the strengths and limitations of the various architectures with respect to predefined criteria.

References

[ABW00] Arroyo, I., Beck, J. E., Woolf, B. P., Beal, C. R., & Schultz, K. (2000). "Microadapting AnimalWatch to gender and cognitive differences with respect to hint interactivity and symbolism." *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, Montreal.

[Ada03]: C. Adams "IGV Taxonomy Development Approach", 2003 Internal document.

[AlM01]: James S. Albus and Alexander M. Meystel *Engineering of Mind: An Introduction to the science of intelligent systems* Wiley Series on Intelligent Systems, 2001.

[And02]: J. R. Anderson *et al.* "An Integrated Theory of the Mind", *Psychological Review.* (2002). pg. 105-172.

[And83]: J.R. Anderson, *The Architecture of Cognition,* Cambridge, MA, Harvard University Press, 1983.

[AnL98]: Anderson, J., & Lebiere, *Atomic Components of Thought*: Lawrence Erlbaum, 1998.

[BCN98]: Bellorini, A., Cacciabue P., Nanetti, N. (1998). Ambiguity and Problems of Communication in Air Traffic Control. Ninth European Conference on Cognitive Ergonomics. 7-15

[KiM96]: Kieras, D.E. and D.E. Meyer *The EPIC Architecture: Principles of Operation*, Technical report, Artificial Intelligence Laboratory, EECS Department, University of Michigan available from http.eecs.umich.edu.people/kieras

[KiM97]: D.E. Kieras and D.E. Meyer "An Overview of the EPIC architecture for cognition and performance with application to human-computer interaction", *Human Computer Interaction*, 12, pp.391-438, 1987.

[Kir96]: Kirschenbaum, S.S., W.D. Gray, R.M. Young "Cognitive Architectures and HCI", SIGCHI, (28)2, April 1996.

[KWA95]: D.A. Kieras, S. D. Wood, K. Abotel, A. Hornof, "GLEAN: A Computer-Based Tool for Rapid GOMS Model Usability Evaluation of User Interface Designs", *UITS* Pittsburg, November 1995. ACM Press.

[LCC]: Laird, Congdon, Coulter *The Soar User's Manual* Version 8.2 Edition 1, Electrical Engineering and Computer Science Department, University of Michigan, Draft of: June 23, 1999.

[Lev87] Leveson, Nancy G. (1987). Technological vs. Human Failure. *The Risks Digest.* Vol.4, No. 94

[Lew99]: Lewis, R.L. (1999) "Cognitive Theory, SOAR", Department of Computer & Information Science The Ohio State University, http://www.eecs.umich.edu/soar.

[LNR87]: Laird, Newell, and Rosenblaum (1987) SOAR: An Architecture for general intelligence, *Artificial Intelligence*, 33, 1-64.

[LPG01] S. Lee, A. Pritchett, D. Goldsman "Hybrid agent-based simulation for analyzing the national airspace system", 33rd Winter Simulation Conference, Arlington, Virginia, 2001, IEEE Computer Society Press, pp.1029-1036.

[New80a]: Newell, A. (1980). Physical symbol systems. *Cognitive Science, 4,* 135-183.

[New80b]: Newell, A. (1980). Reasoning, problem solving and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.), *Attention and Performance VIII* . Hillsdale, NJ: Erlbaum.

[New90]: Newell, A. (1990). *Unified Theories of Cognition.* Cambridge, Massachusetts: Harvard University Press.

[NeS72]: Newell and Simon (1972) *Human Problem Solving* Englewood Cliffs, N.J. Prentice Hall, 1972.

[Nie02]: T.E. Nielson TacAir Soar reference